Markdown-Probleme: Kramdown & XeLaTeX

Um PDFs für meinen Jekyll-Blog mit Markdown zu generieren, verwende ich den folgenden Pandoc-Befehl:

```
command = [
    'pandoc',
    input_markdown_path,
    '-o', output_pdf_path,
    '-f', 'markdown',
    '--pdf-engine', 'xelatex',
    '--resource-path=.:assets',
    '-V', f'CJKmainfont={CJK_FONT}',
    '-V', f'CJKsansfont={CJK_FONT}',
    '-V', f'CJKmonofont={CJK_FONT}',
    '-V', f'geometry:{GEOMETRY}',
    '-V', 'classoption=16pt',
    '-V', 'CJKoptions=Scale=1.1',
    '-V', 'linestretch=1.5'
]
```

Hinweis: Der Code wurde nicht übersetzt, da er in der Programmiersprache Python geschrieben ist und spezifische Befehle enthält, die in ihrer ursprünglichen Form belassen werden sollten, um die Funktionalität zu gewährleisten.

Unterstützung für Kramdown und XeLaTeX

Beim Schreiben von Markdown, das sowohl mit kramdown (für die HTML-Ausgabe von Jekyll) als auch mit XeLaTeX (für die PDF-Ausgabe über Pandoc) funktionieren soll, gibt es einige Dinge zu beachten:

1. Bildpfad-Kompatibilität • Kramdown (HTML): Bevorzugt Pfade, die mit / beginnen, um auf Assets zu verweisen. • XeLaTeX (PDF): Bevorzugt relative Pfade ohne führendes /.

Lösung: Verwenden Sie relative Pfade, die für beide funktionieren:

```
![](assets/images/chatgpt/block.jpg)
```

2. Umgang mit kramdown-Attributen • {:.responsive} ist spezifisch für kramdown und dient der Gestaltung von HTML-Ausgaben. • XeLaTeX unterstützt diese Attribute nicht und wird einen Fehler ausgeben.

Lösung: Entfernen Sie kramdown-spezifische Attribute in Markdown, die für die PDF-Generierung vorgesehen sind. Zum Beispiel:

```
![](assets/images/chatgpt/block.jpg){: .responsive }

![](assets/images/chatgpt/block.jpg)
```

Wenn {:.responsive} für Ihr Jekyll-HTML-Layout entscheidend ist, sollten Sie es selektiv für die Webausgabe hinzufügen, während Sie es im PDF-Generierungsprozess weglassen.

Workflow für Duale Kompatibilität

1. Schreiben Sie Markdown-Inhalte mit minimalen Abhängigkeiten von kramdown-spezifischen Funktionen.
2. Für erweiterte Styling-Optionen in HTML, wenden Sie CSS-Klassen direkt in Ihren Jekyll-Templates an, anstatt sie inline im Markdown zu verwenden.
3. Nutzen Sie Pandoc-Optionen, um die PDF-Formatierung zu steuern, während Sie die Portabilität des Markdowns beibehalten.

Durch die Einhaltung dieser Praktiken bleibt der Markdown-Inhalt sowohl mit der Jekyll-HTML-Darstellung als auch mit der XeLaTeX-PDF-Generierung kompatibel, was einen nahtlosen Workflow für die Veröffentlichung in mehreren Formaten gewährleistet.